

```

1 %% Machine Learning
2 % Lab 4: Regularized Logistic Regression (NonLinear case)
3 % — Microchip Anomaly —
4 %{
5 In this part of the exercise, you will get to try out different regularization
6 parameters for the dataset to understand how regularization prevents overfitting.
7 Notice the changes in the decision boundary as you vary lambda. With a small
8 lambda, you should see that the classifier gets almost every training example
9 correct, but draws a very complicated boundary, thus overfitting the data.
10 This is not a good decision boundary: for example, it predicts
11 that a point at  $x = (-0.25; 1.5)$  is accepted ( $y = 1$ ), which seems to be an
12 incorrect decision given the training set.
13 With a larger lambda, you should see a plot that shows a simpler decision
14 boundary which still separates the positives and negatives fairly well. How-
15 ever, if lambda is set to too high a value, you will not get a good at and the
16 decision
17 boundary will not follow the data so well, thus underfitting the data
18 %}
19 %{
20 % In this part, you are given a dataset with data points that are not
21 % linearly separable. However, you would still like to use logistic
22 % regression to classify the data points.
23 % To do so, you introduce more features to use — in particular, you add
24 % polynomial features to our data matrix (similar to polynomial
25 % regression).
26 %}
27
28 %% Initialization
29 clear ; close all; clc
30
31 %% Load Data
32 % The first two columns contains the X values and the third column
33 % contains the label (y).
34
35 data = load('ex2data2.txt');
36 X = data(:, [1, 2]); Y = data(:, 3);
37
38 plotData(X, Y);
39
40 % Put some labels
41 hold on;
42 % Labels and Legend
43 xlabel('Microchip Test 1')
44 ylabel('Microchip Test 2')
45 % Specified in plot order
46 legend('y = 1', 'y = 0') % Good or Nay
47 hold off;
48
49 %% Add Polynomial Features
50
51 % Note that mapFeature also adds a column of ones for us, so the intercept
52 % term is handled
53 X = mapFeature(X(:,1), X(:,2));
54
55 % Initialize fitting parameters

```

```

56 init_w = zeros(size(X, 2), 1);
57
58 % Set regularization parameter lambda to 1 (you should vary this)
59 % Try the following values of lambda (0, 1, 10, 100).
60 lambda = 0.1;
61
62 %% Set Options
63 options = optimset('GradObj', 'on', 'MaxIter', 400);
64
65 % Optimize
66 [w, J, exit_flag] = ...
67     fminunc(@(t)(costFunctionReg(t, X, Y, lambda)), init_w, options);
68
69 % Plot Boundary
70 plotDecisionBoundary(w, X, Y);
71 hold on;
72 title(sprintf('lambda = %g', lambda))
73
74 % Labels and Legend
75 xlabel('Microchip Test 1')
76 ylabel('Microchip Test 2')
77
78 legend('y = 1', 'y = 0', 'Decision boundary')
79 hold off;
80
81 % Compute accuracy on our training set
82 p = predict(w, X);
83
84 fprintf('Train Accuracy: %f\n', mean(double(p == Y)) * 100);
85 fprintf('Expected accuracy (with lambda = 1): 83.1 (approx)\n');

```

mapFeature.m

```
1 function out = mapFeature(X1, X2)
2 % MAPFEATURE Feature mapping function to polynomial features
3
4 degree=3; % change it to 6 for example
5 out = ones(size(X1(:,1))); % first column ones -> bias
6 for i = 1:degree
7     for j = 0:i
8         out(:, end+1) = (X1.^(i-j)).*(X2.^j);
9     end
10 end
11
12 end
```

sigmoid.m

```
1 function g = sigmoid(z)
2 %SIGMOID Compute sigmoid function
3 g = zeros(size(z));
4 g = 1 ./ (1 + exp(-z));
5 end
```

costFunctionReg.m

```
1 function [C, grad] = costFunctionReg(w, X, Y, lambda)
2 %COSTFUNCTIONREG Compute cost and gradient
3 %for logistic regression with regularization
4
5 m = length(Y); % number of training examples
6 C = 0;
7 grad = zeros(size(w));
8
9 % calculate cost function
10 h = sigmoid(X*w);
11 % calculate penalty
12 % excluded the first w value
13 w1 = [0 ; w(2:size(w), :)];
14 p = lambda*(w1'*w1)/(2*m);
15 % Alternative:
16 %p = (lambda / (2*m))*sum(w_reg.^2);
17
18 C = ((-Y)'*log(h) - (1-Y)'*log(1-h))/m + p;
19 % Alternative:
20 % C = ((1/m).*sum((-1.*Y).*log(hx)) - ((1.-Y).*log(1.-hx)))+p;
21
22 % calculate grads
23 grad = (X'*(h - Y)+lambda*w1)/m;
24
25 end
```